

SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL
FACULDADE DE TECNOLOGIA SENAI/SC FLORIANÓPOLIS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO
DE SISTEMAS

DEREK WILLIAN STAVIS

UMA PROPOSTA DE INTERFACE DO CLIENTE DE TORRENT
TRANSMISSION BASEADO NO GTK HIG 3.14

Florianópolis/SC

2015

DEREK WILLIAN STAVIS

**UMA PROPOSTA DE INTERFACE DO CLIENTE DE TORRENT
TRANSMISSION BASEADO NO GTK HIG 3.14**

Trabalho de Conclusão de Curso apresentado
à Banca Examinadora do Curso Superior de
Tecnologia em Análise e Desenvolvimento de
Sistemas da Faculdade de Tecnologia do SE-
NAI Florianópolis como requisito parcial para
obtenção do Grau de Tecnólogo em Análise e
Desenvolvimento de Sistemas.
Professor Orientador: Prof. Luciana Schmitz,
Esp. (SENAI/SC).

Florianópolis/SC

2015

DEREK WILLIAN STAVIS

**UMA PROPOSTA DE INTERFACE DO CLIENTE DE TORRENT
TRANSMISSION BASEADO NO GTK HIG 3.14**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia do SENAI Florianópolis como requisito parcial para obtenção do Grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

**APROVADA PELA COMISSÃO EXAMINADORA
EM FLORIANÓPOLIS, 18 DE DEZEMBRO DE 2015**

Prof. Luciana Schmitz, Esp. (SENAI/SC)
Coordenador do Curso

Profa. Aline Cristina Antonelli de Oliveira, Esp. (SENAI/SC)
Coordenador de TCC

Prof. Prof. Luciana Schmitz, Esp. (SENAI/SC), Dr. (SENAI/SC)
Orientador

Prof. João Carlos Testi Ferreira, Esp. (SENAI/SC)
Examinador

“A única coisa que não muda é que tudo muda.”
(HERÁCLITO DE ÉFESO)

STAVIS, Derek Willian. **UMA PROPOSTA DE INTERFACE DO CLIENTE DE TORRENT TRANSMISSION BASEADO NO GTK HIG 3.14** Florianópolis, 2015. 32f. Trabalho de Conclusão de Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Faculdade de Tecnologia do SENAI, Florianópolis, 2015.

RESUMO

Os avanços trazidos nos últimos quatro anos no ambiente gráfico GNOME foram de suma importância para atingir uma experiência diferenciada e consistente de uso. Conceitos já difundidos no design de interfaces de usuário foram revisitados, paradigmas foram quebrados, mudando a forma com que os usuários vêem e interagem com o sistema operacional. Entretanto, nem todos os softwares disponíveis para a plataforma acompanharam a velocidade de desenvolvimento da plataforma GNOME, e muitos ainda necessitam de atenção. Com o lançamento do HIG (Guia de Interface com o Usuário) no GNOME versão 3.12 toda filosofia e padrões de design da plataforma foram sintetizados em linguagem simples e objetiva, tornando-se o material referência para o desenvolvimento e manutenção de seu ecossistema de softwares. Esta pesquisa aborda o processo de adaptação do cliente de torrent Transmission de acordo com o guia de interfaces do GNOME versão 3.14.

Palavras-chave: Interfaces Gráficas. GTK+. GNOME. Design. Redesign.

STAVIS, Derek Willian. **UMA PROPOSTA DE INTERFACE DO CLIENTE DE TORRENT TRANSMISSION BASEADO NO GTK HIG 3.14** Florianópolis, 2015. 32f. Trabalho de Conclusão de Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Faculdade de Tecnologia do SENAI, Florianópolis, 2015.

ABSTRACT

The advances in the last four years in GNOME Desktop Environment was utmost to achieve a consistent and distinguished user experience. Widespread concepts already used in user interface design were revisited and paradigms were broken, thus changing how the user see and interact with a computer. Although, not every software available for the GNOME desktop followed with the transition period, and most softwares needs attention. With the release of GNOME Human Interface Guidelines together with version 3.12 of GNOME as a official documentation for design principles, all platform's philosophies and design patterns were translated in a simple and objective language, aiming to be the reference for both development and maintenance of GNOME's ecosystem. This research addresses the thinking process of redesigning the user interface of Transmission, a free and open source client for file sharing, targeting GNOME HIG version 3.14.

Key-words: Graphical User Interfaces. GTK+. GNOME. Design. Redesign.

LISTA DE ILUSTRAÇÕES

Figura 1 – Funcionamento de um gerenciador de janelas	12
Figura 2 – Uma captura de tela do GNOME 3.16	14
Figura 3 – Padrões e suas aplicações	16
Figura 4 – Transmission 2.28 no GNOME 3.16	17
Figura 5 – Medição da razão área útil vs controles	21
Figura 6 – Protótipo funcional do <i>Transmission</i> no GNOME 3.16	23
Figura 7 – Aquisição das razões área útil vs controles	24
Figura 8 – Barra de Ferramentas no <i>Evince</i> e <i>Gedit</i> 3.6	25
Figura 9 – Mapeamento de menu no <i>Evince</i>	26
Figura 10 – Transição da <i>Header Bar</i>	27
Figura 11 – Filtros retráteis através do uso de um Revealer	28
Figura 12 – Transição do menu de limites globais	28
Figura 13 – Ícones simbólicos na barra de estado	29

LISTA DE TABELAS

Tabela 1 – Sistemas operacionais e versões do GNOME	20
Tabela 2 – Área útil vs Controles	24

LISTA DE ABREVIATURAS E SIGLAS

GNU	GNU is Not Unix - Projeto de licenciamento e distribuição de pacotes de software, baseados na licença GNU GPL.
GPL	General Public License - Modelo de licenciamento de software baseado em liberdades.
GTK+	GIMP Toolkit - Biblioteca de componentes para criação de interfaces gráficas.
API	Application-Programmer Interface - Protocolos de comunicação de um determinado pacote de software.
GIMP	GNU Image Manipulation tool - Ferramenta de manipulações de imagem de código aberto, sob licença GNU.
GNOME	Ambiente gráfico de estação de trabalho disponível para GNU/Linux.
HIG	Human Interface Guidelines - Guia de referências de design para implementação de interfaces.

SUMÁRIO

1 INTRODUÇÃO	10
1.1 OBJETIVOS	10
1.1.1 Objetivo geral	10
1.1.2 Objetivos específicos	10
2 REVISÃO DA LITERATURA	11
2.1 Interfaces gráficas com o usuário	11
2.2 Gerenciadores de Janelas	12
2.3 <i>Toolkits Gráficos</i>	13
2.4 O <i>toolkit GTK+</i> e a plataforma GNOME	13
2.5 Software livre e o desenvolvimento contínuo	14
2.6 Impactos na consistência e usabilidade	15
2.7 HIG ou Human Interface Guidelines	15
2.8 Transmission	16
3 PROCEDIMENTOS METODOLÓGICOS	19
3.1 Estudo longitudinal dos padrões de design nas aplicações	19
3.2 Proposição de melhorias na interface do <i>Transmission</i>	21
4 RESULTADOS E DISCUSSÕES	23
4.1 Percepções do processo de transição das aplicações centrais	23
4.2 Processo de redesign da interface do <i>textitTransmission</i>	26
4.2.1 Header Bar	26
4.2.2 Barra de Filtros	27
4.2.3 Limites Globais de Upload/Download	28
4.2.4 Barra de Estado	28
5 CONCLUSÃO	30
REFERÊNCIAS	31

1 INTRODUÇÃO

O GNOME é um ambiente gráfico similar aos apresentados pelo Microsoft Windows e Apple OS X. Com uma experiência de usuário única, o GNOME, diferente de seus similares proprietários, é mantido por uma comunidade de desenvolvedores aberta e sua licença de código-fonte permite que virtualmente qualquer pessoa no mundo possa estudar ou modificar seu funcionamento.

O lançamento de um guia oficial de padrões para o desenvolvimento de interfaces gráficas orientadas ao GNOME foi de suma importância para alinhar desenvolvedores aos objetivos do ambiente gráfico. O guia procura trazer harmonia visual conforme sua implementação, seja através dos aplicativos centrais – os primeiros a receber tratamento estético – ou por aplicativos de terceiros, que devem se adaptar utilizando o guia como referência.

Aplicativos que não se adequam ao guia passam a apresentar-se defasados, perdendo pontos em usabilidade e estética, aumentando a taxa de rejeição. Mantenedores de aplicativos baseados no *toolkit* GTK+ devem, portanto, dedicar atenção para o fator usabilidade, mantendo-se compatível com o progresso da plataforma GNOME.

Um aplicativo que sofre deste efeito e necessita dedicação é o Transmission, um cliente de *torrent* popular, que foi desenvolvido na era GTK+ versão 2, onde os requisitos, recursos e visão da plataforma GNOME eram diferentes (BENSON; CLARK; NICKELL, 2014).

1.1 OBJETIVOS

1.1.1 Objetivo geral

Propor adaptações na interface gráfica GTK+ do cliente de torrent Transmission através da análise de casos de uso de aplicativos centrais da plataforma GNOME, versão 3.16, e de acordo com as recomendações do HIG versão 3.14, trazendo mais harmonia para seus utilizadores na plataforma GNOME.

1.1.2 Objetivos específicos

- (a) Comparar dados evolutivos da interface de aplicações centrais do GNOME.
- (b) Identificar pontos de redesign na janela principal do Transmission.
- (c) Propor adaptações na interface do Transmission.

2 REVISÃO DA LITERATURA

O desenvolvimento de mecanismos que facilitam tarefas existe desde a idade das pedras, onde hominídeos produziam ferramentas para auxílio próprio. A transversalidade do conhecimento e da experimentação nos levou a descoberta de novas metodologias e o aperfeiçoamento de técnicas, consequentemente modificando a forma com que o ser humano interage com o mundo.

2.1 Interfaces gráficas com o usuário

O conceito de interface é extremamente amplo, e foi largamente difundido com o início dos estudos de interação humano-máquina. O avanço da industrialização permitiu com que máquinas extremamente complexas substituíssem seres humanos nas tarefas mais difíceis, deixando estes, seus operadores, apenas com a responsabilidade de pilotá-las de forma simples e segura.

O advento de tecnologias multimídias interativas como computadores, celulares e *tablets* elevou o patamar da criação de interfaces e gestão de tarefas ao estado da arte, agregando conhecimentos transversais de artistas visuais a músicos.

Ferreira, Barr e Noble (2005, p. 1) afirmam que a criação de “interfaces com o usuário ainda está mais para arte do que para ciência.”, e justificam, explicando que “A maior parte do design ou redesign é baseado em estudos empíricos ou protótipos, e ainda há muito pouca compreensão teórica ou de engenharia de como conduzir o processo de design e produzir bons designs pela primeira vez”.

Interfaces gráficas modernas foram alcançadas através da associação entre *hardware* e *software*, podendo utilizar de um ou mais dispositivos de entrada e saída, com o intuito de promover usabilidade e fácil adaptação.

Dispositivos de entrada transformam coordenadas do mundo real para o mundo virtual, registrando condições externas que podem ser modificadas através da interação com um ou mais atores, os usuários. Dispositivos de entrada comuns são mouse, tela de toque, teclado, etc.

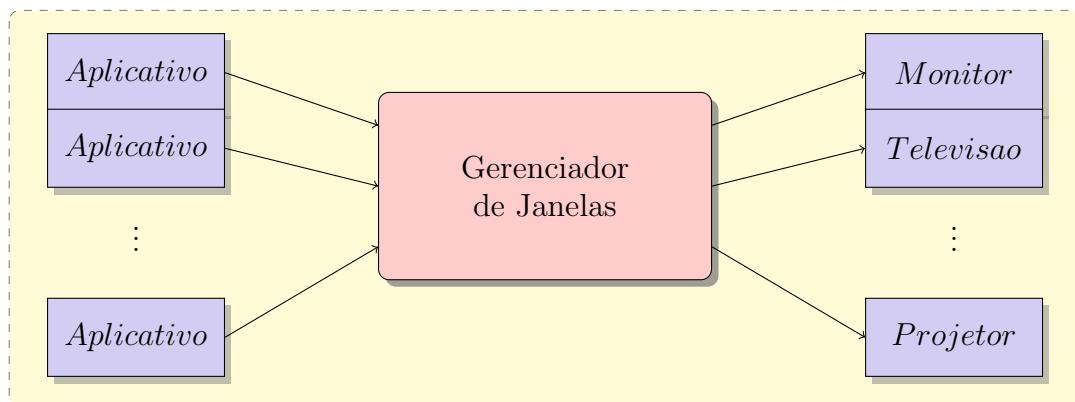
Dispositivos de saída projetam informações geradas por um sistema computacional e seu caráter é geralmente baseado nos sentidos: Visão, audição, tato. O dispositivo de saída mais comum é o monitor, que tem por finalidade projetar imagens compatíveis com a capacidade humana de visão.

2.2 Gerenciadores de Janelas

Gerenciadores de janelas tem como principal função dividir a imagem de um dispositivo de saída (geralmente monitores) em múltiplas regiões de desenho, denominadas janelas (MYERS, 1996, p. 5).

O primeiro conceito acadêmico de gerenciamento de tarefas através da sobreposição de janelas data de 1969, na tese de Ph.D. de Alan Kay. A implementação de seu conceito, vista pela primeira vez em funcionamento no sistema do Xerox PARC, é largamente utilizada até hoje por grandes sistemas tanto comerciais quanto de código aberto, como Windows, OS X, GNOME, KDE (MYERS; HUDSON; PAUSCH, 2000, p. 7).

Figura 1 – Funcionamento de um gerenciador de janelas



Fonte: Do Autor (2015)

Um gerenciador de janelas moderno também tem por finalidade coordenar a exibição de um conjunto de janelas em um conjunto de monitores (conforme fluxograma da Figura 1), escutar por eventos de entrada (mouse, teclado) e informar os responsáveis pelas janelas sobre alterações no layout de tela (dimensões da tela, dimensões da janela, espaço de cor).

Gerenciadores de janelas, porém, não tem por responsabilidade preencher as áreas de desenho com gráficos, e como suas *Application Programming Interfaces* (APIs) operam geralmente a nível de pixel, a tarefa de escrever um programa gráfico acaba sendo demorada e entediante. Além disso, se cada desenvolvedor criasse seus próprios componentes, seria praticamente impossível disponibilizar uma experiência consistente ao usuário (ROSENTHAL, 1988 apud MYERS; HUDSON; PAUSCH, 2000).

Através das abstrações disponibilizadas pelos gerenciadores de janelas múltiplas aplicações gráficas podem coexistir em um mesmo monitor. A construção deste tipo de aplicação não é de responsabilidade do gerenciador de janelas, e para isso ferramentas conhecidas como *toolkits* foram criadas.

2.3 *Toolkits* Gráficos

A principal finalidade de um *toolkit* é viabilizar a interação humano- computador, permitindo a manipulação direta de elementos gráficos metafóricos, chamados widgets. Widgets são os componentes funcionais essenciais da interface gráfica e através deles o usuário pode interagir com programas de computador.

Com seus próprios conjuntos de widgets, *toolkits* possibilitam a criação de programas com alta consistência visual e comportamental, através da aplicação de estilos e/ou comportamento próprios nos componentes de interface gráfica com o usuário (MYERS; HUDSON; PAUSCH, 2000).

As responsabilidades de um *toolkit* vão de desenhar até processar eventos de um ou mais dispositivos de entrada (mouse, teclado, painel de toque), verificando a colisão de um evento com um widget (clique em um botão, por exemplo), traduzindo e informando os eventos para a aplicação proprietária.

Do ponto de vista do programador a finalidade de um *toolkit* é abstrair características de baixo nível, promovendo a reutilização de código, e amortizando o custo de desenvolvimento, reduzindo o tempo de desenvolvimento de novos projetos (HAEFLIGER; KROGH; SPAETH, 2008).

Toolkits multiplataforma podem ser utilizados para escrever interfaces gráficas portáveis, permitindo com que o mesmo código seja compilado para um sistema operacional diferente do em que foi escrito e funcione da mesma forma.

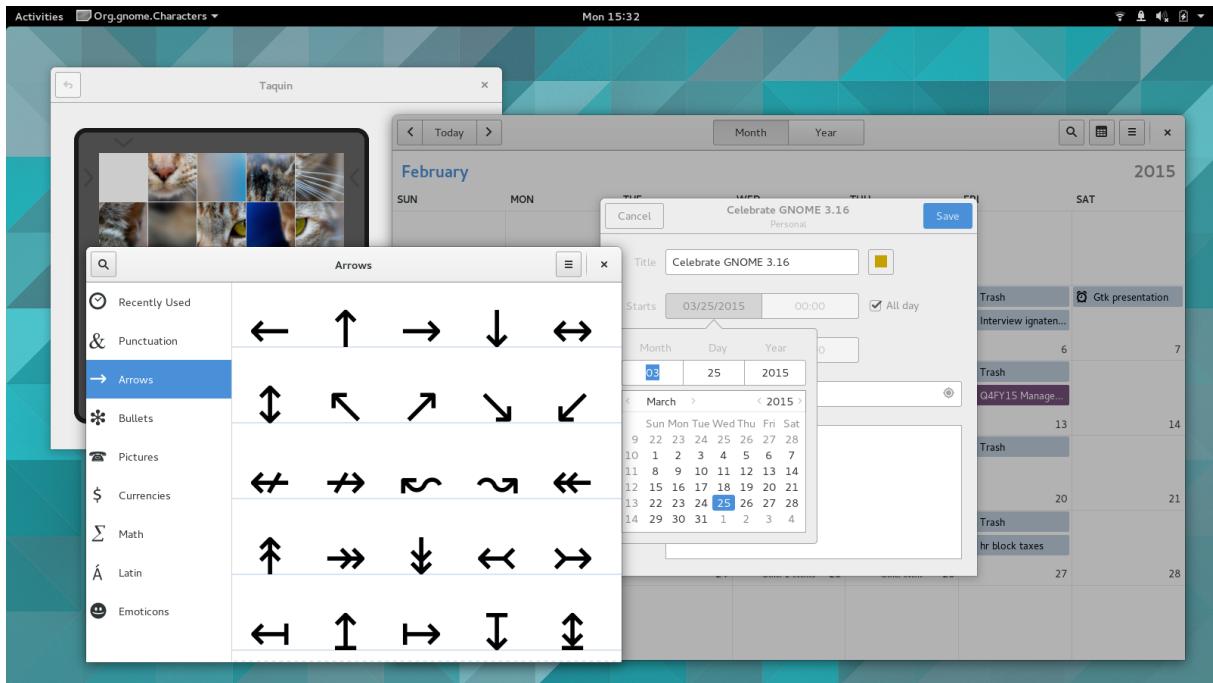
2.4 O *toolkit* GTK+ e a plataforma GNOME

Existem diversas opções de gerenciadores de janela de código aberto, comumente incluídos em distribuições de Linux. O GNOME, um ambiente gráfico bastante difundido pelos usuários de Linux, foi fundado e está em ativo desenvolvimento por uma comunidade de engenheiros de software ao redor do mundo.

O GNOME 3 é uma maneira fácil e elegante de usar seu computador. Ele foi desenhado para lhe colocar no controle e trazer liberdade para todos. O GNOME 3 é desenvolvido pela comunidade GNOME, um grupo internacional e diverso de contribuidores que são suportados por uma fundação independente, e sem fins lucrativos (THE GNOME PROJECT, 2015).

A experiência GNOME é construída por um conjunto de aplicações centrais, que incluem um gerenciador de janelas, um lançador de aplicações, calculadora, editor de texto, gerenciadores de arquivos, contatos, etc.

Figura 2 – Uma captura de tela do GNOME 3.16



Fonte: (CLASEN, 2015)

A plataforma GNOME e seus aplicativos centrais são baseados no *toolkit* GTK+. O GTK+ é formado por um conjunto de ferramentas multi-plataforma para criar interfaces gráficas de usuário. Por oferecer um conjunto completo de widgets é adequado para projetos desde pequenas ferramentas pontuais até suítes completas de aplicativos (THE GTK+ PROJECT, 2015).

2.5 Software livre e o desenvolvimento contínuo

Uma das características das plataformas de código aberto é a distribuição de esforços em prol do constante desenvolvimento e melhoria. A pluralidade de opiniões e ideias eleva o patamar das discussões e permite com que vários pontos de vista sejam levados em consideração na evolução da plataforma.

Apesar dos prós existentes na distribuição de esforços também existem os contras – Projetos que são desenvolvidos paralelamente nem sempre avançam na mesma velocidade. O contra fica mais sério quando um projeto depende do outro, como é o caso de *toolkits* e programas que consomem suas APIs.

Quando ocorrem mudanças na interface de programação de uma framework ou biblioteca, softwares dependentes tem de se adaptar as mudanças. O efeito cascata provocado pela propagação de alterações na malha de softwares dependentes não é incomum, e já foi objeto de estudo (YAU; COLLOFELLO; MACGREGOR, 1978).

2.6 Impactos na consistência e usabilidade

Além dos aplicativos centrais, geralmente garantidos de acompanhar a evolução da plataforma – sua ergonomia e visual – existe uma vasta gama de aplicações tanto de código aberto quanto proprietárias disponíveis para atender as mais variadas necessidades.

A grande maioria das aplicações GTK+ são mantidas pela comunidade de software livre, e se não atualizadas podem ficar defasadas em usabilidade, ergonomia e consistência com a plataforma. Shneiderman (1992) descreve a usabilidade como uma combinação das seguintes características:

- Facilidade de aprendizado
- Alta velocidade de operação
- Baixa taxa de erros
- Satisfação do usuário
- Retenção de usuários pelo tempo

Em prol da usabilidade a maioria dos *toolkits* aplica um modelo próprio de apresentação, organização, interação e estilização de widgets, atendendo aos requisitos do tipo de ambiente onde é executado (desktop, tablet e celular).

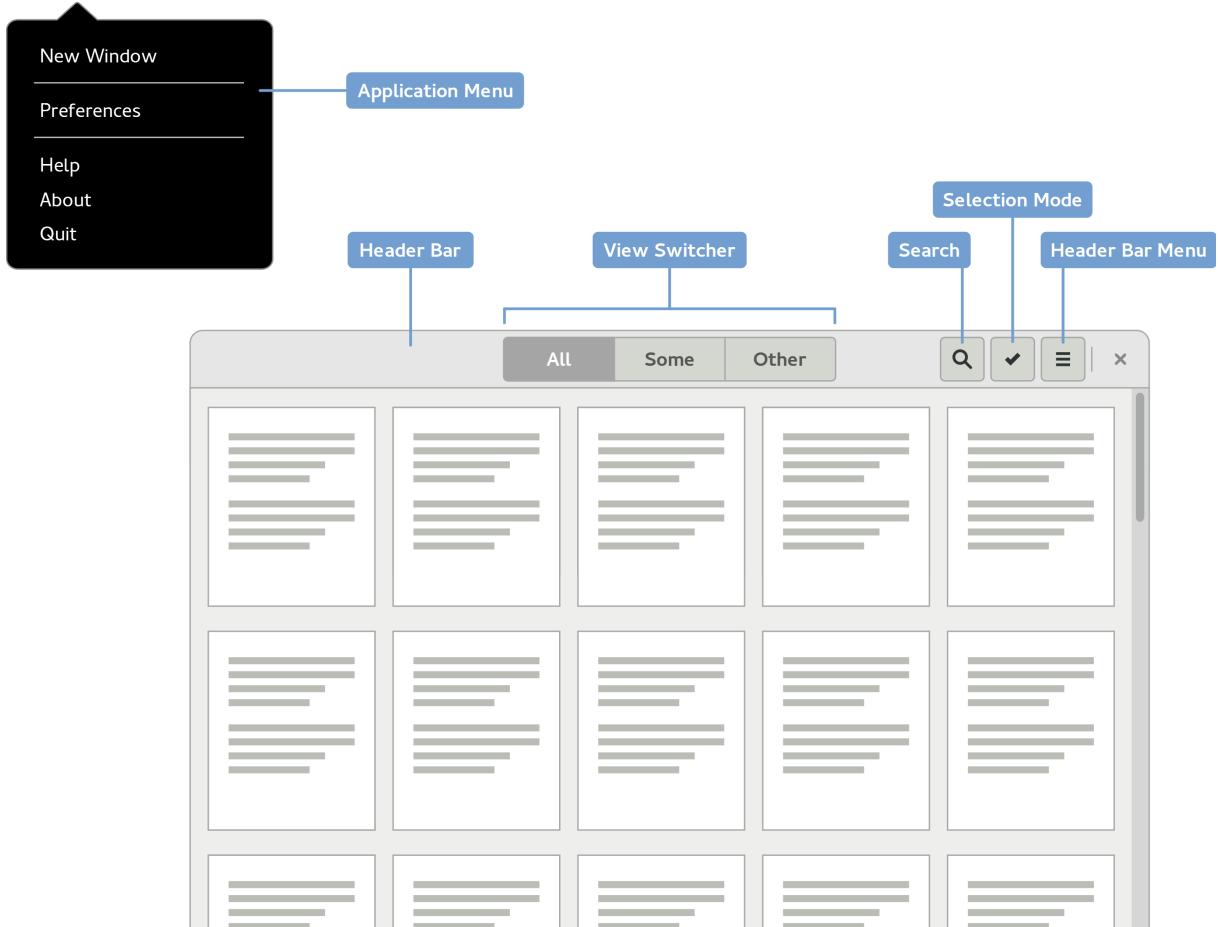
2.7 HIG ou Human Interface Guidelines

Divulgado no ano de 2014, acompanhando a versão 3.14 da plataforma GNOME, sob o título de *Human Interface Guidelines*, o conjunto de padrões de design oficialmente promovido pela GNOME Foundation busca promover a máxima integração de interfaces gráficas GTK+ na plataforma GNOME.

Se você é um desenvolvedor com experiência limitada de design o HIG foi planejado para auxiliar você a criar facilmente uma interface de usuário efetiva. Para designers, o HIG provê uma introdução as possibilidades ao usar o GTK+, assim como padrões de design que são usados nos aplicativos GNOME (DAY, 2014a).

O HIG, como também é chamado, é uma literatura ilustrada de diretrizes recomendadas no desenvolvimento de interfaces gráficas que utilizem o *toolkit*, com o intuito de reforçar a consistência visual e integração com o ambiente GNOME. A figura 3 ilustra alguns dos padrões de design documentados pelo HIG.

Figura 3 – Padrões e suas aplicações



Fonte: (DAY, 2014b)

Dentre os objetivos do HIG estão a transmissão das metas de design de alto nível e estratégicas da experiência GNOME e a comunicação das diretrizes de design essenciais, de uma forma clara e viva, acompanhando a evolução da plataforma.

2.8 Transmission

Um aplicativo famoso disponível para a plataforma GNOME é o cliente de *BitTorrent* Transmission (The Transmission Project, 2013). Aclamado pela sua simplicidade, o aplicativo é utilizado para compartilhar arquivos através da Internet.

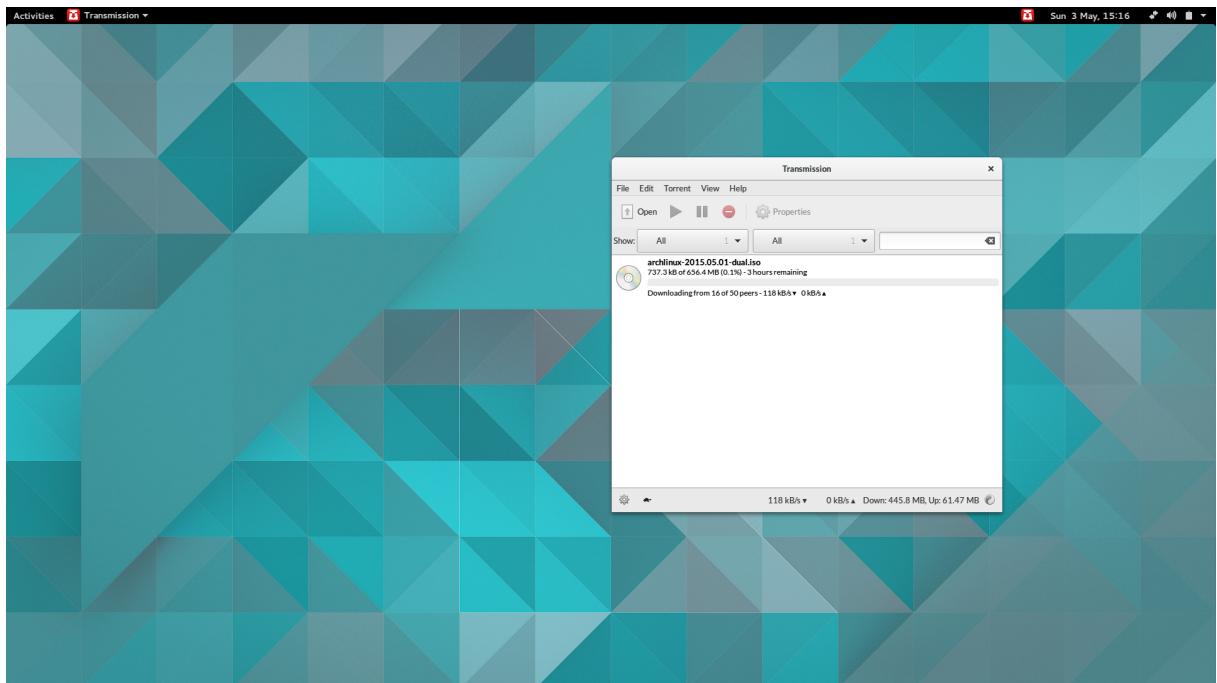
A funcionalidade básica de um cliente de torrent como o Transmission é interpretar um arquivo que representa uma transferência (chamado de torrent), e transferir os arquivos nele contidos.

Transferências do tipo torrent dividem arquivos em múltiplas partes, paralelizando o processo ao transferir de múltiplos pontos, além de serem de via dupla – partes já recebidas também são enviadas para outros clientes interessados no arquivo.

Utilizando-se de uma configuração padrão funcional e poucos cliques para configurar funcionalidades avançadas a utilização do cliente de torrent Transmission foi projetada para ser fácil e poderosa (THE TRANSMISSION PROJECT, 2015).

A primeira versão do Transmission foi lançada em Setembro de 2005, já com uma interface gráfica baseada em GTK+, projetada sob as recomendações do HIG da época, publicada digitalmente em formato de livro (BENSON; CLARK; NICKELL, 2014).

Figura 4 – Transmission 2.28 no GNOME 3.16



Fonte: Do Autor (2015)

Como pode ser visto na Figura 4, a janela principal do Transmission é composta por uma barra de menus sensível a seleção e uma lista de todas transferências, onde cada ítem exibe suas informações básicas, atualizadas constantemente: Nome, progresso, quantidade de pontos, velocidade de recebimento, velocidade de envio. Além de exibir todas transferências, o Transmission também permite filtrá-las por estado: Recebendo, enviando, pausada, etc.

Como as transferências podem consumir grande quantidade de dados, outros aplicativos que necessitam de rede podem ser afetados, por exemplo: Uma página da web pode demorar mais para carregar. Para isso o Transmission permite impor limites de velocidade no recebimento e envio, de forma global – qual é utilizada o tempo todo pelo cliente, ou temporariamente – qual pode ser ativada e desativada ao clique de um botão.

Funcionalidades avançadas incluem pausar transferências ao atingir uma determinada razão de recebimento/envio, monitorar diretórios por arquivo torrent e adicioná-los

como transferências automaticamente, verificar as condições de rede necessárias para uma performance ótima, dentre outras.

3 PROCEDIMENTOS METODOLÓGICOS

A presente pesquisa aplicada foi desenvolvida através da exploração de casos de uso tanto da evolução de aplicativos centrais da plataforma GNOME quanto padrões de design utilizados em outras versões do cliente de torrent *Transmission*.

Segundo Jung e Eng (2003), o objetivo da pesquisa aplicada é gerar inovação frente a uma demanda ou necessidade, e seu desenvolvimento consiste na utilização dos conhecimentos adquiridos na pesquisa básica associados a pesquisa tecnológica, com a finalidade de obter aplicações práticas.

O produto final desta pesquisa se concretiza na forma de um protótipo funcional de interface gráfica compatível com os padrões definidos no HIG 3.14 para o cliente de torrent *Transmission*.

3.1 Estudo longitudinal dos padrões de design nas aplicações

Através de pesquisa exploratória com o objetivo de avaliar o processo de evolução da plataforma GNOME e seu design, foram identificadas aplicações centrais do GNOME 3.16 que já implementam os padrões de design definidos pelo HIG (DAY, 2014a).

Jung e Eng (2003) caracteriza a pesquisa exploratória pela extração de conhecimento sobre um dado fenômeno, sem grande teorização, visando descobrir práticas e diretrizes alternativas ao conhecimento existente.

Foram escolhidas três aplicações centrais a serem analisadas. Pelo papel significativo desempenhado na experiência GNOME foram selecionadas as seguintes aplicações:

- Nautilus – Navegador de Arquivos¹
- Evince – Visualizador de Documentos Digitais²
- Gedit – Editor de Arquivos de Texto³

Com propósito de avaliar a dinâmica do processo de design do GNOME ao longo do tempo, um estudo de caráter longitudinal foi elaborado nas três aplicações escolhidas, através da comparação de duas versões distantes do GNOME e das aplicações centrais escolhidas.

¹ Disponível em <https://wiki.gnome.org/Apps/Nautilus>, acesso em Maio de 2015

² Disponível em <https://wiki.gnome.org/Apps/Evince>, acesso em Maio de 2015

³ Disponível em <https://wiki.gnome.org/Apps/Gedit>, acesso em Maio de 2015

A montagem do ambiente necessário para a execução do estudo longitudinal foi facilitada pela distribuição oficial de sistemas operacionais com todo software necessário para uma experiência GNOME básica, prontos para usar através da transferência para uma mídia inicializável⁴. Detalhadas na Tabela 1 estão as informações das versões escolhidas, dentre as disponíveis para estudo.

A soma de verificação de cada arquivo foi utilizada para checar a integridade do sistema operacional, e todas aplicações foram analisadas na versão oficial – sem alterações na configuração do sistema, troca de temas, fontes, etc.

Tabela 1 – Sistemas operacionais e versões do GNOME

Versão do GNOME	Data de Modificação	Nome do arquivo	Soma de Verificação
3.6.0	08/10/2012	GNOME-3.6.0.iso	MD5 - 753c99ce2342f658 65c1f74bc3722e44
3.16	25/03/2015	gnome-3.16.x86-64.iso	SHA256 - 4a6185a0aca89f15 8f769d76d5a0086f 0f1e9d709a5d80cd cf2b0d52d67ab2b2

Fonte: Do Autor (2015)

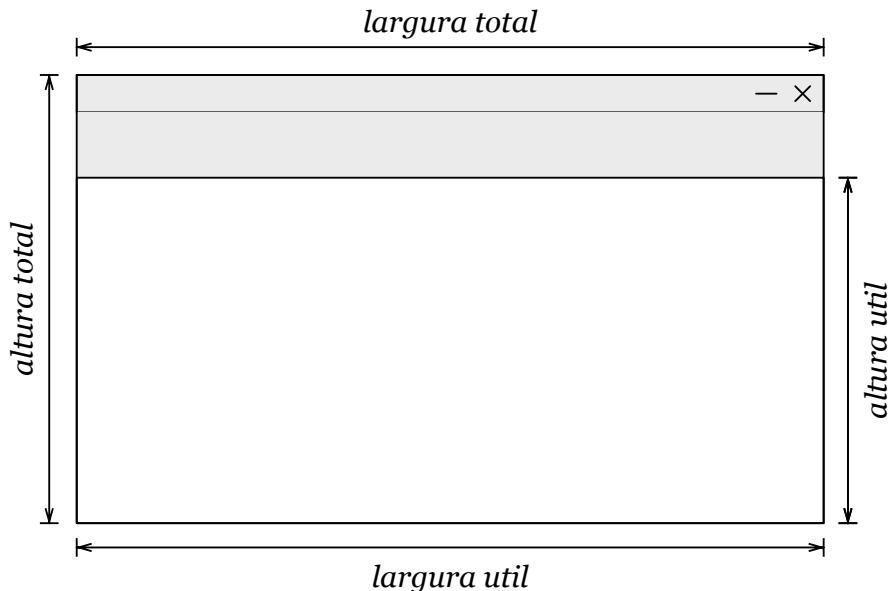
Os padrões de design incluídos na análise foram os especificados pelo HIG (DAY, 2014b), e as variáveis observadas no estudo incluíram, porém não se limitaram a (i) área útil vs controles, onde foi analisada a razão entre a quantidade de pixels de área útil versus controles da aplicação e (ii) relocação de funcionalidade, onde foram analisadas mudanças operacionais em um ou mais elementos de interface, sendo estas de caráter quantitativo e qualitativo, respectivamente.

A obtenção da razão área útil versus controles foi feita na janela principal de cada aplicativo através da fórmula $razao = areacontroles / areautil$. Considera-se *areautil* a área do aplicativo na qual a informação principal é apresentada, enquanto a *areacontroles* pode ser obtida através da subtração da *areautil* do tamanho total da janela, em pixels, conforme exemplificado na Figura 5. No caso do *Nutilus*, a lista de arquivos e barra lateral formam a área principal, no *Evince*, o documento apresentado, e no caso do *Gedit*, a área de edição de texto (incluindo as abas de arquivos abertos).

A análise de relocação de funcionalidades foi feita através da correlação de um elemento de interface existente na versão mais antiga com um elemento correspondente na versão mais recente. Por exemplo, a funcionalidade de um menu, na versão mais antiga, encontrado na forma de um botão na versão mais recente.

⁴ Disponível em <https://download.gnome.org/misc/promo-usb>, acesso em Maio de 2015

Figura 5 – Medição da razão área útil vs controles



Fonte: Do Autor (2015)

3.2 Proposição de melhorias na interface do *Transmission*

Beer et al. (2009) afirmam que a prototipação de interfaces é uma técnica de evolução de design que tem como base a constante interação e avaliação dos resultados. Bäumer et al. (1996, p. 2) classifica e separa em quatro classes os protótipos de interfaces gráficas:

- (a) De apresentação
- (b) Funcional
- (c) Experimental
- (d) Piloto

Pela (i) maleabilidade de implementar estrategicamente ambas interface gráfica e funcionalidade de um programa, com o propósito de averiguar o funcionamento de um conceito e (ii) pelo fato do *Transmission* permitir alterações em seu código- fonte através de sua licença GPL o protótipo funcional foi escolhido como método experimental para a geração de propostas de melhorias na interface do *Transmission*.

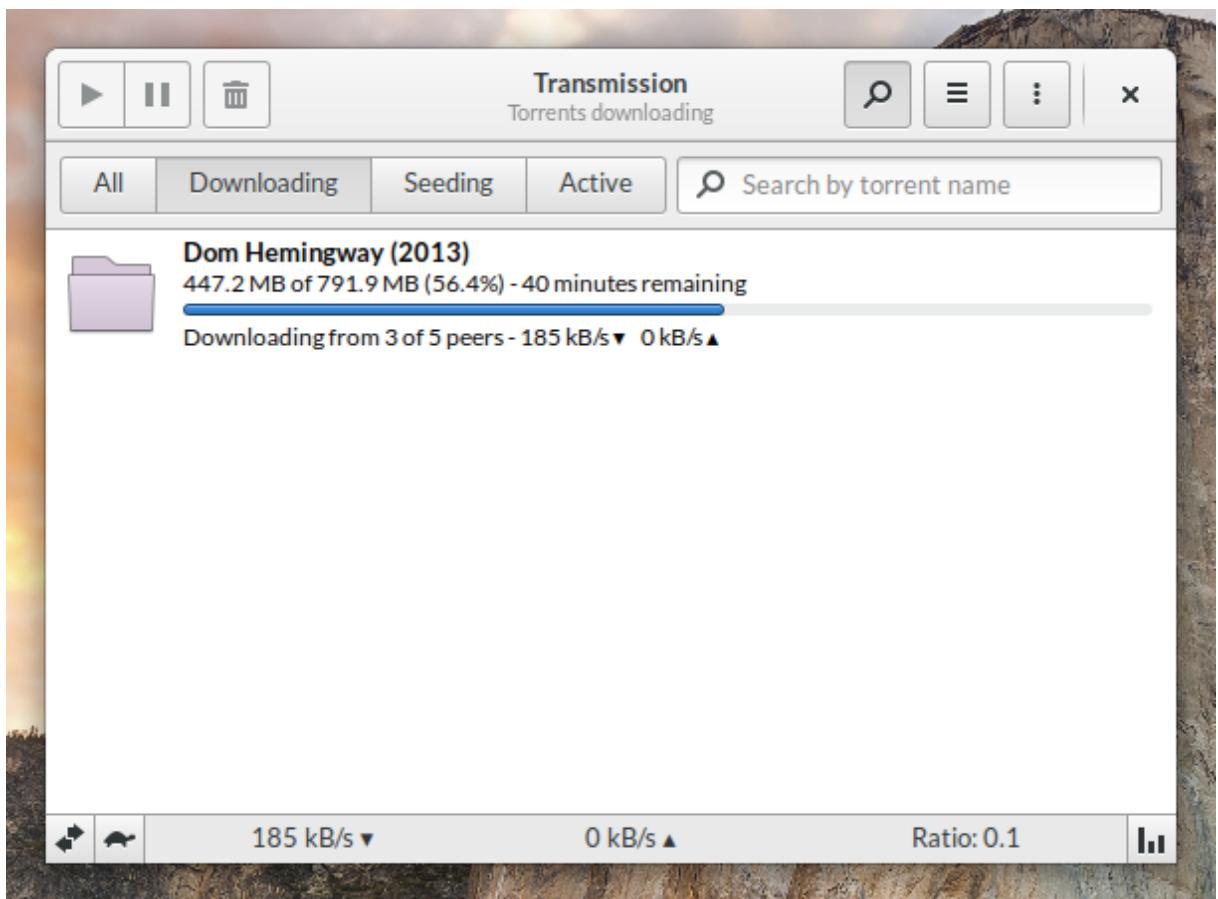
O protótipo funcional foi gerado através de alterações experimentais no código- fonte do *Transmission* objetivando modificar sua interface gráfica utilizando como base as

tendências encontradas no estudo longitudinal das aplicações centrais e também as diretrivas estabelecidas no HIG.

4 RESULTADOS E DISCUSSÕES

A Figura 6 apresenta o protótipo funcional resultado da aplicação desta pesquisa, embasado na análise das aplicações centrais e estudo do HIG, gerado através da alteração do código-fonte do *Transmission* versão 2.82.

Figura 6 – Protótipo funcional do *Transmission* no GNOME 3.16



Fonte: Do Autor (2015)

Verificou-se que o embasamento nos dados evolutivos das aplicações centrais do GNOME foi de grande utilidade para compreender o processo de redesign dos aplicativos desta plataforma, possibilitando e facilitando a geração das propostas de redesign do *Transmission*.

4.1 Percepções do processo de transição das aplicações centrais

Durante o processo de análise longitudinal percebeu-se uma tendência clara de redução da razão área útil vs controles em todas aplicações centrais analisadas. Também

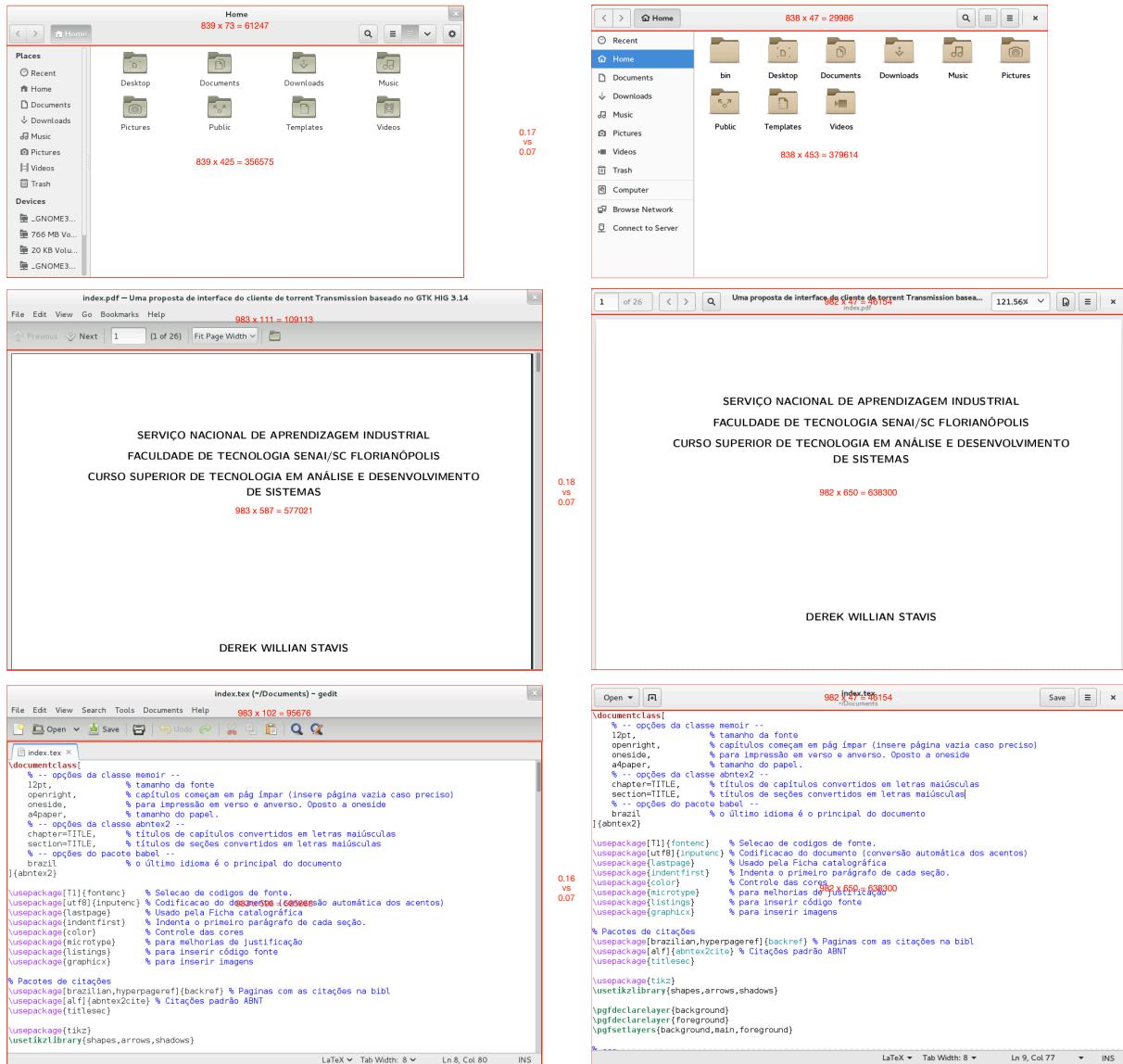
foi possível perceber a padronização da razão na versão 3.16 de todas as aplicações. A Tabela 2 apresenta as razões área útil vs controles calculadas, enquanto a figura Figura 7 ilustra o processo de aquisição.

Tabela 2 – Área útil vs Controles

Aplicação	Versão 3.6	Versão 3.16
<i>Nautilus</i>	0.17	0.07
<i>Evince</i>	0.18	0.07
<i>Gedit</i>	0.16	0.07

Fonte: Do Autor (2015)

Figura 7 – Aquisição das razões área útil vs controles



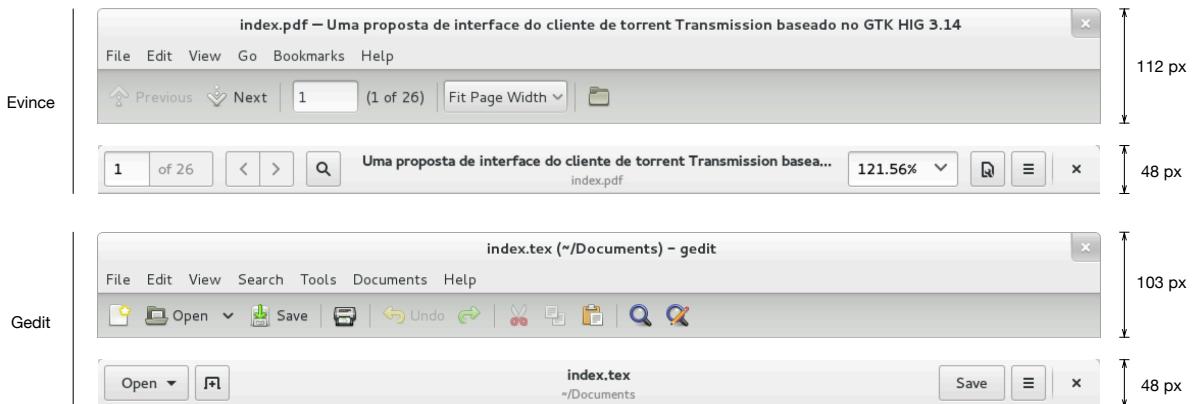
Fonte: Do Autor (2015)

A tendência de redução desta razão se repete na remodelagem da forma e função de vários widgets, uma clara influência de ambos design de usabilidade e da redução de espaço vertical, visto a padronização de resoluções baseados na razão 16:9 (720p, 1080p, etc).

Falando especificamente sobre os elementos de interface, um dos elementos mais comuns de um ambiente gráfico baseado em janelas é a barra de título, por onde o usuário pode mover a janela no espaço de trabalho. Até então, no GNOME, desenvolvedores tinham pouco ou nenhum controle sobre elas por questões de compatibilidade de software.

Com a introdução do conceito de *Header Bar*, as antigas barras de título passam a acomodar botões, caixas de texto, sliders, etc, substituindo as barras de ferramentas, antes encontradas comumente abaixo de menus ou das barras de título, conforme a Figura 8.

Figura 8 – Barra de Ferramentas no *Evince* e *Gedit* 3.6

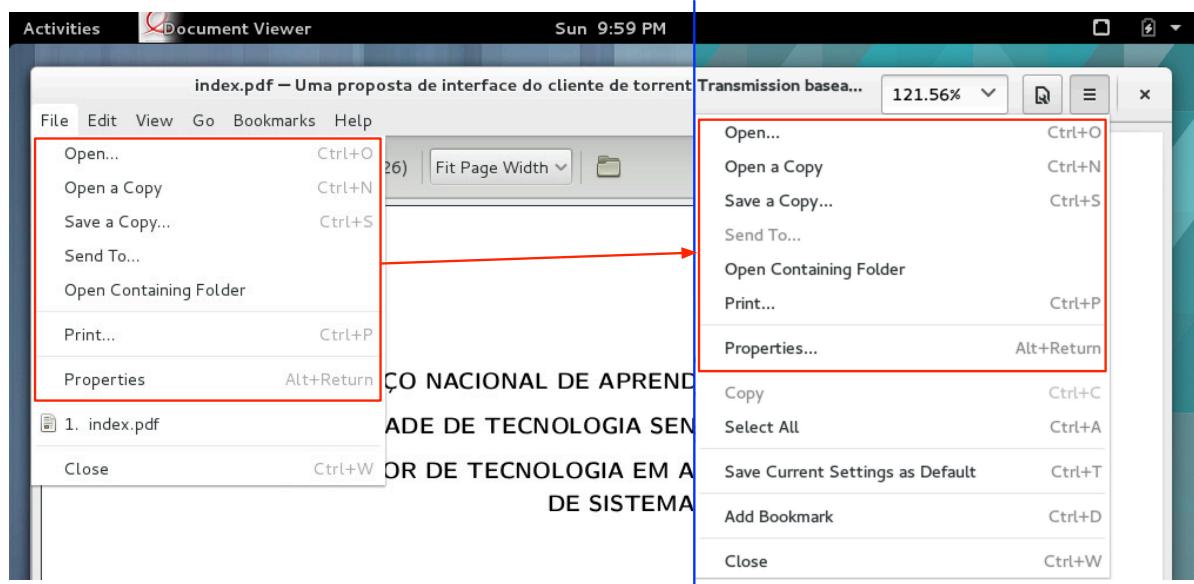


Fonte: Do Autor (2015)

Constatou-se que, dentre as aplicações analisadas na versão 3.6, o *Nautilus* foi a mais avançada em relação aos padrões de design definidos pelo HIG, antes mesmo deste guia ser publicado, através da pré-implementação de uma *Header Bar*, abaixo da barra de título, evidenciando a experimentação progressiva dos novos padrões antes de uma aplicação geral.

Outra mudança notável foi a transição da barra de menus, quais ações foram dissolvidas através de um ou mais widgets na janela do programa (sendo grande maioria na *Header Bar*), nas ações pertinentes a janela, e em um menu de aplicação, para ações pertinentes ao programa (Abrir, Sobre, Sair). A Figura 9 apresenta um mapeamento aproximado das ações da barra de menus do *Evince* 3.6 para o 3.16.

A utilização do *Revealer* em ambos *Nautilus* e *Evince* também demonstrou-se eficaz como técnica para redução de espaço vertical e aumento da área útil de ambos programas, através da exibição temporária da caixa de busca, alternável pelo botão lupa no *Header*

Figura 9 – Mapeamento de menu no *Evince*

Fonte: Do Autor (2015)

Bar.

O *Popover*, um elemento de interface introduzido no HIG 3.12, é uma visão alternável e flutuante que permite adicionar menus, listas e widgets variados em seu interior, e é comumente usado como parte de menus ou menus de contexto. Sua utilização foi observada em diversos pontos, inclusive como coadjuvante na transição da barra de menus em todas aplicações analisadas.

4.2 Processo de redesign da interface do textitTransmission

Em relação a utilização do *Transmission* após o contato com as aplicações centrais do GNOME 3.16 fica claro, logo na primeira utilização, que sua interface não adere aos padrões incluídos no HIG, e se mostra completamente desatualizada.

Foi possível, através da análise da transição das aplicações centrais do GNOME, compilar uma série de modificações que podem ser aplicadas em prol da melhoria da compatibilidade da interface do *Transmission* com a experiência GNOME, quais serão apresentadas a seguir.

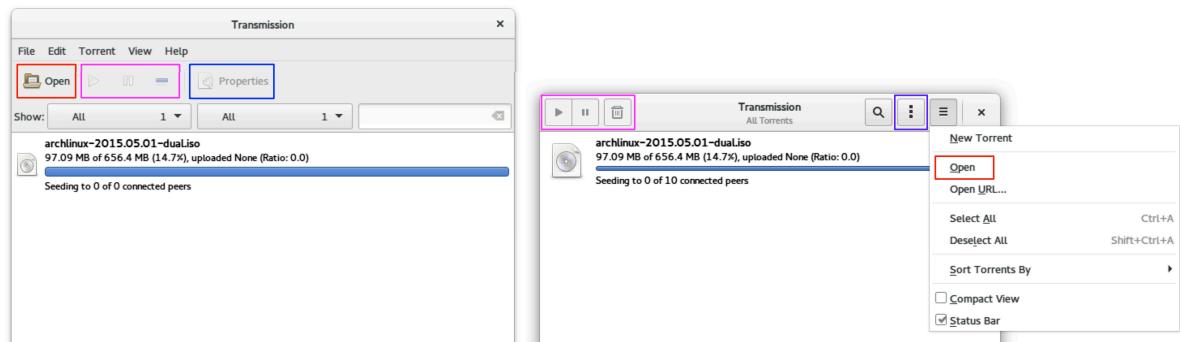
4.2.1 Header Bar

A implementação da *Header Bar* no *Transmission* foi efetivada através da dissolução das ações encontradas em ambas barra de ferramentas e menus em botões. As ações da

barra de menus foram separadas em duas categorias de menu: Ações pertinentes ao programa e ações pertinentes a transferência selecionada, posicionados da direita para a esquerda, respectivamente, conforme indicado pelo HIG.

A Figura 10 apresenta um paralelo entre a interface antiga e a interface proposta, relacionando através de marcação em cores as transições efetuadas. Os botões sensíveis a seleção (Iniciar, Parar e Remover) puderam ser mantidos com seu comportamento original.

Figura 10 – Transição da *Header Bar*



Fonte: Do Autor (2015)

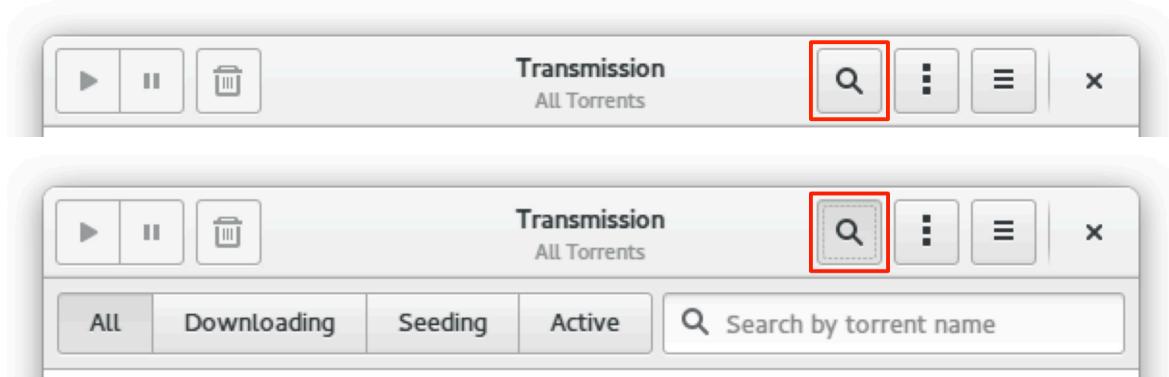
Pela natureza compacta da janela do *Transmission* e pelo compartilhamento do espaço disponível na *Header Bar* pelos botões de ação e título da aplicação, a adição excessiva de botões pode suprimir o título da janela, e a presença de um número diferente de botões no lados direito e esquerdo pode fazer parecer com que o título não esteja centralizado, causando desconforto visual. Por este motivo os botões “Properties” e “Open” foram migrados para os menus, sendo este movido para o menu de janela e aquele para o menu de seleção.

4.2.2 Barra de Filtros

Seguindo a implementação utilizada no *Evince* e *Nautilus*, os filtros de atividade foram movidos juntamente a caixa de busca textual para um *Revealer*, ativado através do botão com ícone de lupa, destacado na Figura 11.

Na versão original do *Transmission*, além do filtro textual, existem dois tipos de filtros adicionais: Por estado (Recebendo, Transmitindo, Parado) e por *tracker* (Servidores que auxiliam a troca de arquivos). Por questões de poupança de espaço a funcionalidade de filtragem por *tracker* foi deliberadamente removida.

Figura 11 – Filtros retráteis através do uso de um Revealer

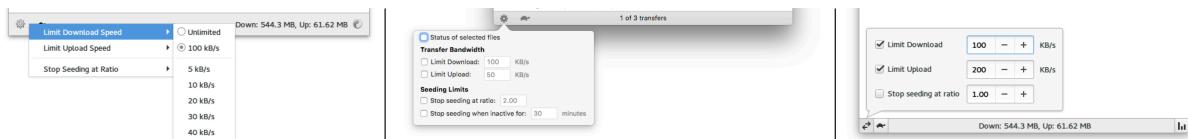


Fonte: Do Autor (2015)

4.2.3 Limites Globais de Upload/Download

Utilizando-se de um padrão de design já existente na interface gráfica do *Transmission* para Apple Mac OS X como fonte de inspiração, foi implementada a configuração de limite de velocidade global utilizando um *Popover*, substituindo o extenso menu existente anteriormente, de forma a permitir uma configuração fácil e rápida com poucos cliques. A Figura 12 apresenta um paralelo entre a implementação original, a fonte de inspiração e o resultado derivado.

Figura 12 – Transição do menu de limites globais

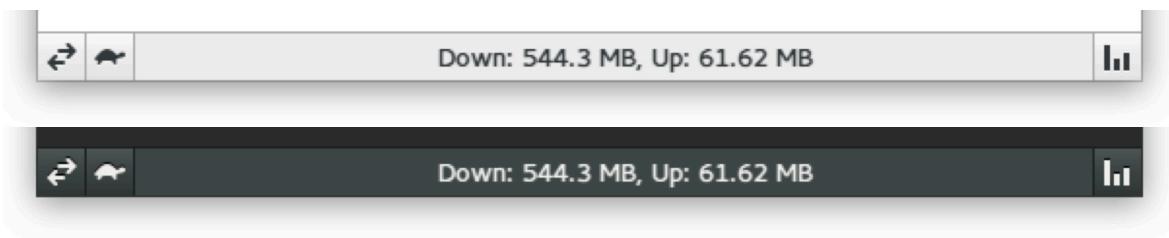


Fonte: Do Autor (2015)

4.2.4 Barra de Estado

Inspirada na barra de estado do *Gedit*, de altura compacta, e utilizando-se das facilidades do GTK+ 3, todos os botões da barra de estado foram implementados utilizando ícones simbólicos, de forma a se adaptarem a temas claros e escuros (ilustrado na Figura 13). Para implementar as opções de visualização das estatísticas (último botão da esquerda para a direita) foi utilizado um *Popover*.

Figura 13 – Ícones simbólicos na barra de estado



Fonte: Do Autor (2015)

5 CONCLUSÃO

Os resultados obtidos através desta pesquisa foram de grande satisfatoriedade, tanto em relação ao seu processo de desenvolvimento, qual trouxe conhecimentos valiosos, cumprindo com todos seus objetivos, quanto em relação ao resultado em si, qual trouxe maior harmonia para a utilização do Transmission na plataforma GNOME.

Em relação a sua titulação, o autor acredita que o embasamento nas aplicações centrais que já implementam os padrões definidos pelo HIG fez com que o redesign esteja de acordo com este guia.

Como perspectiva de trabalhos futuros, o presente autor espera que a motivação desta pesquisa seja inspiração para o redesign de outras aplicações que necessitem da mesma atenção, na expectativa de que este trabalho contribua positivamente para o desenvolvimento de trabalhos futuros.

REFERÊNCIAS

- BÄUMER, D. et al. User interface prototyping—concepts, tools, and experience. In: IEEE COMPUTER SOCIETY. *Proceedings of the 18th international conference on Software engineering*. [S.l.], 1996. p. 532–541. Citado na página 21.
- BEER, D. J. D. et al. Client-centred design evolution via functional prototyping. *International Journal of Product Development*, Inderscience, v. 8, n. 1, p. 22–41, 2009. Citado na página 21.
- BENSON, C.; CLARK, B.; NICKELL, S. *GNOME Human Interface Guidelines*. 2014. GNOME Human Interface Guidelines 2.2.1. Disponível em: <<https://developer.gnome.org/hig-book/2.32>>. Acesso em: 5 apr. 2015. Citado 2 vezes nas páginas 10 e 17.
- CLASEN, M. *GNOME 3.16 SIGHTINGS*. 2015. GNOME 3.16 SIGHTINGS. Disponível em: <<https://blogs.gnome.org/mclasen/2015/02/23/gnome-3-16-sightings>>. Acesso em: 10 apr. 2015. Citado na página 14.
- DAY, A. *GNOME Human Interface Guidelines*. 2014. GNOME Human Interface Guidelines. Disponível em: <<https://developer.gnome.org/hig/3.14>>. Acesso em: 5 apr. 2015. Citado 2 vezes nas páginas 15 e 19.
- DAY, A. *Patterns*. 2014. GNOME Human Interface Guidelines. Disponível em: <<https://developer.gnome.org/hig/3.14>>. Acesso em: 5 apr. 2015. Citado 2 vezes nas páginas 16 e 20.
- FERREIRA, J.; BARR, P.; NOBLE, J. The semiotics of user interface redesign. In: AUSTRALIAN COMPUTER SOCIETY, INC. *Proceedings of the Sixth Australasian conference on User interface-Volume 40*. [S.l.], 2005. p. 47–53. Citado na página 11.
- HAEFLIGER, S.; KROGH, G. V.; SPAETH, S. Code reuse in open source software. *Management Science*, INFORMS, v. 54, n. 1, p. 180–193, 2008. Citado na página 13.
- JUNG, C. F.; ENG, M. Metodologia científica. *Ênfase em pesquisa tecnológica*, v. 3, 2003. Citado na página 19.
- MYERS, B.; HUDSON, S. E.; PAUSCH, R. Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction (TOCHI)*, ACM, v. 7, n. 1, p. 3–28, 2000. Citado 2 vezes nas páginas 12 e 13.
- MYERS, B. A. Uimss, toolkits, interface builders1. 1996. Citado na página 12.
- ROSENTHAL, D. A simple x11 client program—or-how hard can it really be to write "hello, world"? In: *USENIX Winter*. [S.l.: s.n.], 1988. p. 229–242. Citado na página 12.
- SHNEIDERMAN, B. *Designing the user interface: strategies for effective human-computer interaction*. [S.l.]: Addison-Wesley Reading, MA, 1992. Citado na página 15.
- THE GNOME PROJECT. *GNOME*. 2015. GNOME. Disponível em: <<https://gnome.org>>. Acesso em: 8 apr. 2015. Citado na página 13.

THE GTK+ PROJECT. *The GTK+ Project*. 2015. The GTK+ Project. Disponível em: <<http://gtk.org>>. Acesso em: 10 apr. 2015. Citado na página 14.

The Transmission Project. *Transmission*. 2013. Disponível em: <<http://transmissionbt.com>>. Citado na página 16.

THE TRANSMISSION PROJECT. *About Transmission*. 2015. About. Disponível em: <<http://www.transmissionbt.com/about/>>. Acesso em: 3 apr. 2015. Citado na página 17.

YAU, S. S.; COLLOFELLO, J. S.; MACGREGOR, T. Ripple effect analysis of software maintenance. In: *Computer Software and Applications Conference, 1978. COMPSAC'78. The IEEE Computer Society's Second International*. [S.l.: s.n.], 1978. p. 60–65. Citado na página 14.